

In the Claims

1(Currently Amended). A text to XML transformer, comprising:

a transformer program having a plurality of ~~executable statements~~ compound statement; and

a processor for executing the transformer program and converting an input text document into an XML document wherein the XML document does not contain every element that was in the input text.

2(Original). The transformer of claim 1, wherein the text document is a structured text document.

3(Original). The transformer of claim 1, wherein the text document is a semi-structured text document.

4(Original). The transformer of claim 1, wherein the input text document has at least two formats.

5(Original). The transformer of claim 4, wherein the text to XML commands include a field separator command that defines a field separator in the text document.

6(Original). The transformer of claim 5, wherein the field separator is a comma.

7(Original). The transformer of claim 5, wherein the field separator is a regular expression.

8(Previously Presented). The transformer of claim 4, wherein a text to XML commands include a match command that requires a field in the input text document to match a character string or a record is skipped.

3/4

44

Do Not Enter - G.V. 10/16/06

```

46  <?xml version="1.0"?>
48  <tx:transform xmlns:tx="http://www.xaware.com/2003/01/text-to-xml"
  version="1.0">
  <!-- initialize and set the document element -->
  50  <tx:template match="BEGIN">
    51  <tx:variable name="FS" value="","","/>
      <addresses>
        <tx:continue/>  52
      </addresses>
    </tx:template>
    <!-- skip the first record because it is just a header -->
    54  <tx:template match="NR = 1">
      56  <tx:next/>
      </tx:template>
      <!-- add address for category customers -->
      58  <tx:template match="F[7] = 'customers'">
        62  <address name=" {F[2]}">
          64  <name>
            66  <tx:value-of expr="F[1]"/><tx:text> </tx:text>
            68  <tx:value-of expr="F[2]"/>
          70  </name>
          72  <street><tx:value-of expr="F[3]"/></street>
            <city><tx:value-of expr="F[4]"/></city>
            <state><tx:value-of expr="F[5]"/></state>
            <zip><tx:value-of expr="F[6]"/></zip>
          </address>
        </tx:template>
      </tx:transform>

```

**FIG. 5**80

[App Settings]  
 window=maximized  
 advanced mode=true  
 install=C:\someapp

[Tool Settings]  
 tool1 driver = lib\somedll.dll /s=46 /t="fast"

**FIG. 6**